



Entwicklungsumgebungen mit Vagrant

CommitterConf 2015, Essen

11. November 2015



Christian Berendt
Cloud Solution Architect
B1 Systems GmbH
berendt@b1-systems.de

Vorstellung B1 Systems

- gegründet 2004
- primär Linux/Open Source-Themen
- national & international tätig
- über 70 Mitarbeiter
- unabhängig von Soft- und Hardware-Herstellern
- Leistungsangebot:
 - Beratung & Consulting
 - Support
 - Entwicklung
 - Training
 - Betrieb
 - Lösungen
- dezentrale Strukturen

Schwerpunkte

- Virtualisierung (XEN, KVM & RHEV)
- Systemmanagement (Spacewalk, Red Hat Satellite, SUSE Manager)
- Konfigurationsmanagement (Puppet & Chef)
- Monitoring (Nagios & Icinga)
- IaaS Cloud (OpenStack & SUSE Cloud & RDO)
- Hochverfügbarkeit (Pacemaker)
- Shared Storage (GPFS, OCFS2, DRBD & CEPH)
- Dateiaustausch (ownCloud)
- Paketierung (Open Build Service)
- Administratoren oder Entwickler zur Unterstützung des Teams vor Ort



Vagrant Designziele 1/2

Schafft Entwicklungsumgebungen, die:

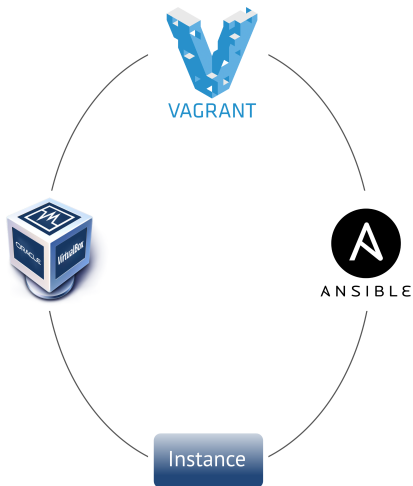
- *schlank*
- *reproduzierbar und*
- *portabel*

sind!

Vagrant Designziele 2/2

- Portabilität (z. B. lokale Entwicklungsumgebungen in der Cloud verwenden)
- Einheitlichkeit (z. B. eine Basisumgebung für alles erstellen)
- Wiederverwendbarkeit (z. B. in einem Continuous Integration System)
- Reproduzierbarkeit
- beschleunigte Entwicklung (z. B. früher Zugriff für UX Designer)

Vagrant Workflow



Vagrant Fakten

- viele Community Plugins
 - <https://github.com/mitchellh/vagrant/wiki/Available-Vagrant-Plugins>
- zahlreiche vorgefertigte Machine Images:
 - <https://atlas.hashicorp.com/boxes/search>
 - packer zum Bauen der Machine Images
- riesige Wissensbasis
 - <http://stackoverflow.com/search?q=vagrant>
- in Ruby geschrieben
- Quellen unter <https://github.com/mitchellh/vagrant>

Vagrant Installation

- <https://www.vagrantup.com/downloads.html>
 - Linux (RPM) – openSUSE, Fedora, ...
 - Linux (DEB) – Ubuntu, Debian, ...
 - Mac OS X
 - Windows

```
$ vagrant version
Installed Version: 1.7.2
Latest Version: 1.7.2
```

You're running an up-to-date version of Vagrant!

Vagrant Konfiguration 1/2

- Vagrantfile
- <http://docs.vagrantup.com/v2/vagrantfile/index.html>

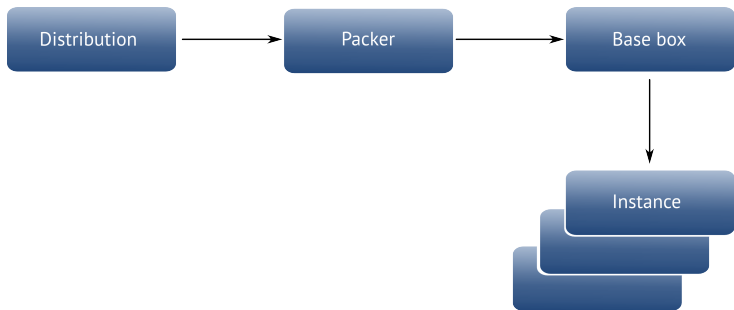
```
Vagrant.configure(2) do |config|  
  # configuration ..  
end
```

Vagrant Konfiguration 2/2

- http://docs.vagrantup.com/v2/vagrantfile/machine_settings.html

```
Vagrant.configure(2) do |config|
  config.vm.define 'testing' do |node|
    node.vm.hostname = 'testing'
  end
end
```

Vagrant Boxes 1/5






Vagrant Boxes 2/5

Discover Vagrant Boxes

This page lets you discover and use Vagrant Boxes created by the community. You can search by operating system, architecture or provider.

Provider filter

Sort by

 ubuntu/trusty64 Official Ubuntu Server 14.04 LTS (Trusty Tahr) builds	2,962,500 downloads 20150430.00 last release 14 hours ago
 hashicorp/precise64 A standard Ubuntu 12.04 LTS 64-bit box.	1,920,177 downloads 1.1.0 last release 1 years ago
 laravel/homestead Official Laravel local development box.	1,343,211 downloads 0.2.6 last release 6 days ago

<https://atlas.hashicorp.com/boxes/search>

Vagrant Boxes 3/5

- <http://docs.vagrantup.com/v2/boxes.html>

```
Vagrant.configure(2) do |config|
  config.vm.define 'node' do |node|
    node.vm.hostname = 'testing'
    node.vm.box = 'ubuntu/trusty64'
  end
end
```

Vagrant Boxes 4/5

- Neue Box hinzufügen: `vagrant box add`

```
$ vagrant box add ubuntu/trusty64
==> box: Loading metadata for box 'ubuntu/trusty64'
      box: URL: https://atlas.hashicorp.com/ubuntu/trusty64
==> box: Adding box 'ubuntu/trusty64' (v20150430.0.0) for provider: virtualbox
      box: Downloading: https://atlas.hashicorp.com/ubuntu/boxes/trusty64/versions
            /20150430.0.0/providers/virtualbox.box
==> box: Successfully added box 'ubuntu/trusty64' (v20150430.0.0) for 'virtualbox'!
```

Vagrant Boxes 5/5

- Liste aller Boxes ausgeben: `vagrant box list`

```
$ vagrant box list
b1-systems/opensuse (virtualbox, 13.2)
boxcutter/centos71  (virtualbox, 1.0.15)
ubuntu/trusty64    (virtualbox, 20150430.0.0)
```


Vagrant Standard Provider

- Docker
- Hyper-V
- VMware Fusion & Workstation
- VirtualBox

<https://docs.vagrantup.com/v2/providers/index.html>

Vagrant Bootstrap 1/4

- Bootstrap mit vagrant up mit dem virtualbox Provider

```
$ vagrant up
Bringing machine 'testing' up with 'virtualbox' provider...
==> testing: Importing base box 'ubuntu/trusty64'...
==> testing: Matching MAC address for NAT networking...
==> testing: Checking if box 'ubuntu/trusty64' is up to date...
==> testing: Setting the name of the VM: testing_testing_1430999179693_84021
[...]
==> testing: Machine booted and ready!
==> testing: Checking for guest additions in VM...
==> testing: Setting hostname...
==> testing: Mounting shared folders...
testing: /vagrant => /home/berendt/testing
```

Vagrant Bootstrap 2/4

- Status anzeigen: `vagrant status`

```
$ vagrant status
Current machine states:

testing                               running (virtualbox)
[...]
```

Vagrant Bootstrap 3/4

- Konsolenzugriff: `vagrant ssh`

```
$ vagrant ssh
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-52-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Thu May  7 11:50:08 UTC 2015

System load:  0.92                Processes:            85
Usage of /:   2.8% of 39.34GB     Users logged in:    0
Memory usage: 29%                IP address for eth0: 10.0.2.15
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

vagrant@testing:~$
```

Vagrant Bootstrap 4/4

- „Aufräumen“: `vagrant destroy`

```
$ vagrant destroy
  testing: Are you sure you want to destroy the 'testing' VM? [y/N] y
==> testing: Forcing shutdown of VM...
==> testing: Destroying VM and associated drives...
```

Vagrant Standard Provisioner

- Ansible
- Shell/File
- Chef
- CFEngine
- Docker
- Puppet
- Salt

<https://docs.vagrantup.com/v2/provisioning/index.html>

Shell Provisioner 1/3

```
node.vm.provision "shell" do |shell|
  shell.inline = "echo hello vagrant"
end
```

Shell Provisioner 2/3

```
$ vagrant up
[...]
==> testing: Running provisioner: shell...
    testing: Running: inline script
==> testing: stdin: is not a tty
==> testing: hello vagrant
```


Shell Provisioner 3/3

```
==> testing: stdin: is not a tty
```

```
config.ssh.shell = "bash -c 'BASH_ENV=/etc/profile exec bash'"
```

Ansible Provisioner 1/2

```
---  
- hosts: all  
  sudo: True  
  tasks:  
    - apt: name=apache2 state=latest  
    - service: name=apache2 enabled=yes state=started
```

<https://github.com/ansible/ansible>

Ansible Provisioner 2/2

```
$ vagrant up
[...]
PLAY [all] *****

GATHERING FACTS *****
ok: [testing]

TASK: [apt name=apache2 state=latest] *****
changed: [testing]

TASK: [service name=apache2 enabled=yes state=started]
ok: [testing]

PLAY RECAP *****
testing: ok=3    changed=1    unreachable=0    failed=0
```

Vagrant – Weitere Features 1/4

- Port Forwarding vom Host System

```
config.vm.network "forwarded_port"
```

- http://docs.vagrantup.com/v2/networking/forwarded_ports.html

Vagrant – Weitere Features 2/4

- synchronisierte Ordner

```
config.vm.synced_folder "src/", "/srv/website"
```

- http://docs.vagrantup.com/v2/synced-folders/basic_usage.html

Vagrant – Weitere Features 3/4

- weitere private Netzwerke verwenden

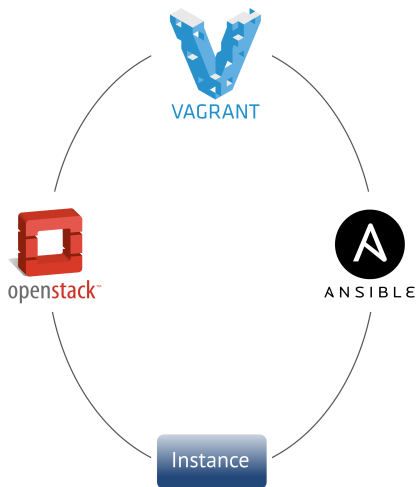
```
config.vm.network "private_network", ip: "192.168.50.4"
```

- http://docs.vagrantup.com/v2/networking/private_network.html

Vagrant – Weitere Features 4/4

- zusätzliche Block Storage Devices hinzufügen
- Multisystem-Umgebungen

Vagrant Workflow mit OpenStack



OpenStack Provider

- mitchellh/vagrant-rackspace
 - **giamarchi/vagrant-openstack-provider**
 - cloudbau/vagrant-openstack-plugin

OpenStack Provider – Installation

- <https://github.com/ggiamarchi/vagrant-openstack-provider>

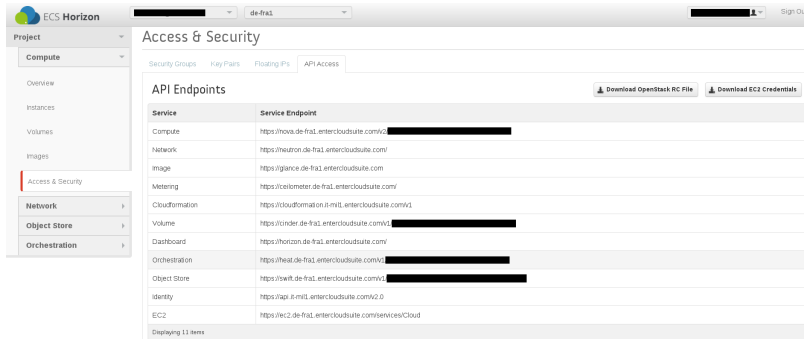
```
$ vagrant plugin install vagrant-openstack-provider
Installing the 'vagrant-openstack-provider' plugin. This can take a few minutes...
Installed the plugin 'vagrant-openstack-provider (0.6.1)'
```

OpenStack Provider – Konfiguration

- SSH Benutzernamen angeben, abhängig vom verwendeten Cloud Image

```
require 'vagrant-openstack-provider'  
Vagrant.configure('2') do |config|  
  config.ssh.username = 'ubuntu'  
  config.vm.provider :openstack do |os|  
    os.server_name = 'vagrant'  
  end  
end  
end
```

OpenStack provider – Credentials 1/2



The screenshot shows the ECS Horizon interface for a project named 'de-fra1'. The 'Access & Security' section is active, displaying a list of API Endpoints. Two buttons are visible: 'Download OpenStack RC File' and 'Download EC2 Credentials'.

Service	Service Endpoint
Compute	https://nova.de-fra1-entercloudsuite.com/v2/ [REDACTED]
Network	https://neutron.de-fra1-entercloudsuite.com/
Image	https://glance.de-fra1-entercloudsuite.com
Metering	https://ceilometer.de-fra1-entercloudsuite.com/
Cloudformation	https://cloudformation.i-mlt-entercloudsuite.com/v1
Volume	https://cinder.de-fra1-entercloudsuite.com/v1 [REDACTED]
Dashboard	https://horizon.de-fra1-entercloudsuite.com/
Orchestration	https://heat.de-fra1-entercloudsuite.com/v1 [REDACTED]
Object Store	https://swift.de-fra1-entercloudsuite.com/v1 [REDACTED]
Identity	https://api.i-mlt-entercloudsuite.com/v2.0
EC2	https://ec2.de-fra1-entercloudsuite.com/services/Cloud

Displaying 13 items

OpenStack Provider – Credentials 2/2

- Einträge in der Datei openrc:

```
export OS_AUTH_URL=https://api.de-fra1.entercloudsuite.com/v2.0
export OS_TENANT_NAME="christian@berendt.io"
export OS_USERNAME="christian@berendt.io"
export OS_PASSWORD="password"
```

- Konfigurationsabschnitt in Vagrantfile
- **Wichtig:** /tokens zu os.openstack_auth_url hinzufügen

```
config.vm.provider :openstack do |os|
  os.openstack_auth_url = 'https://api.de-fra1.entercloudsuite.com/v2.0/tokens'
  os.username           = 'christian@berendt.io'
  os.password           = 'password'
  os.tenant_name        = 'christian@berendt.io'
end
```

OpenStack Provider – Ressourcen

- Kommandos zum Auflisten existierender Cloud-Ressourcen:
 - Flavor – `flavor-list`
 - Floating IP Adresse – `floatingip-list`
 - Image – `image-list`
 - Interne Netzwerke – `subnet-list`

OpenStack Provider – Flavors

- verfügbare Flavors auflisten: `vagrant openstack flavor-list`

Id	Name	vCPU	RAM (Mo)	Disk size (Go)
10	e1standard.x1	1	512	20
20	e1standard.x2	1	1024	40
210	e2standard.x1	1	512	0
220	e2standard.x2	1	1024	0
230	e2standard.x3	2	2048	0
240	e2standard.x4	2	4096	0
250	e2standard.x5	4	8192	0
260	e2standard.x6	6	15360	0
270	e2standard.x7	8	30720	0
280	e2standard.x8	16	102400	0
30	e1standard.x3	2	2048	80
40	e1standard.x4	2	4096	160
50	e1standard.x5	4	8192	320
60	e1standard.x6	6	15360	620
70	e1standard.x7	8	30720	1200

- diese Session verwendet `e1standard.x1`

OpenStack Provider – Floating IPs

- verfügbare Floating IP Pools auflisten: `vagrant openstack floatingip-list`

```
+-----+
| Floating IP pools |
+-----+
| PublicNetwork    |
+-----+

+----+----+-----+-----+
| Id | IP | Pool | Instance id |
+----+----+-----+-----+
+----+----+-----+-----+
```

- diese Session verwendet `PublicNetwork`

OpenStack Provider – Images

- alle verfügbaren Images auflisten: `vagrant openstack image-list`

Id	Name
17e9c7e5-1ce8-4bb8-b9bb-2b8314d0bbb0b	GNU/Linux Ubuntu Server 14.10 Utopic Unicorn [Daily Build 20150509] x64 RAW
848280af-2419-404d-9585-44c1a2100b0f	GNU/Linux Ubuntu Server 14.10 Utopic Unicorn [Daily Build 20150509] x64
570097e8-3bc7-4168-b066-4fce7284d7b4	GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr [Daily Build 20150512] x64 RAW
dbf3fe7f-689a-4871-bb49-62d2de986551	GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr [Daily Build 20150512] x64
0a19bf95-edf6-4174-b36e-589dbc381170	GNU/Linux CoreOS 668.3.0
0ba389b2-eb57-4086-943d-8cc79b16f649	BSD FreeBSD 10.0 Production x64
fce6d24b-935c-40e3-ab84-7f8d2317d0d4	GNU/Linux RHEL 7
c80633a3-2331-4074-8bed-8ca113e001d9	Microsoft Windows Server 2008 R2 Datacenter x64 RAW
3a06a09f-55b4-45c0-a9a3-55363d06d083	Microsoft Windows Server 2012 Datacenter x64 RAW
bbd61ef8-21a9-4811-95ab-b0fc140ca983	Microsoft Windows Server 2012 R2 Datacenter x64 RAW
24f1be7a-6080-462d-a7df-0b484958fba6	Microsoft Windows Server 2012 Datacenter x64
d2caa91b-f560-423f-b131-df6738004b746	Microsoft Windows Server 2012 R2 Datacenter x64
82146de1-775e-4b0a-977d-2f6738004b746	OwnCloud7
44bef40d-f465-4c8b-b186-1f062190a72f	GNU/Linux Ubuntu Server 12.04 LTS Precise Pangolin [Daily Build 20141212] RAW x64
74c478f8-57ee-4df1-bdb3-d777166847d5	GNU/Linux Ubuntu Server 12.04 LTS Precise Pangolin [Daily Build 20141212] x64
70a0165a-371b-4dc6-a63d-b0ee28970e43	GNU/Linux CentOS 6.5 x64
5e52c201-5cee-4bb0-95e2-c2e87b33409e	Microsoft Windows 10 Technical Preview x64
2a03edd5-15c8-41df-98c0-0cfdcd4bf04d5	Microsoft Windows 8.1 Pro x64
728ad0c4-6adc-448d-9c3c-be174be351d6	GNU/Linux CentOS 7 RAW x64
10894243-0caa-477f-81bd-f306dedb2126	GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64
f9c665ae-7ef2-4739-8a0d-879128e8d12c	GNU/Linux CirrOS 0.3.2 x64
3bbe2b54-4545-4b42-936e-7ccba4324b74	GNU/Linux Debian 7.4 Wheezy x64
f9e70348-478b-4620-bcec-4be5619c5b5c	GNU/Linux Fedora 20 Heisenbug x64
65475456-bbe1-40cb-9eca-54bc70315387	GNU/Linux CentOS 5.10 x64
28436bfb-cdbf-4709-842c-f5482e8b10de	GNU/Linux Debian 6.0 Squeeze x64

- diese Session verwendet GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64

OpenStack Provider – Konfiguration

- OpenStack Ressourcen in Vagrantfile festlegen
 - Flavor: e1standard.x1
 - Floating IP Pool: PublicNetwork
 - Image: GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64

```
config.vm.provider :openstack do |os|  
  os.openstack_auth_url = 'https://api.de-fra1.entercloudsuite.com/v2.0/tokens'  
  os.username           = 'christian@berendt.io'  
  os.password           = 'password'  
  os.tenant_name        = 'christian@berendt.io'  
  os.flavor              = 'e1standard.x1'  
  os.floating_ip_pool   = 'PublicNetwork'  
  os.image               = 'GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64'  
end
```

OpenStack Provider – Bootstrap, erster Versuch

```
$ vagrant up
Bringing machine 'default' up with 'openstack' provider...
==> default: Finding flavor for server...
==> default: Finding image for server...
==> default: Launching a server with the following settings...
==> default: -- Tenant      : christian@berendt.io
==> default: -- Name       : default
==> default: -- Flavor     : e1standard.x1
==> default: -- FlavorRef  : 10
==> default: -- Image      : GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64
==> default: -- ImageRef   : f0094243-0caa-4f7f-81bd-f306dedb2126
==> default: -- KeyPair    : vagrant-generated-8k637omz
{"forbidden": {"message": "It is not allowed to create an interface on external
network 50ea4b59-42e9-4427-9f00-16362fd0cfd9", "code": 403}}
```

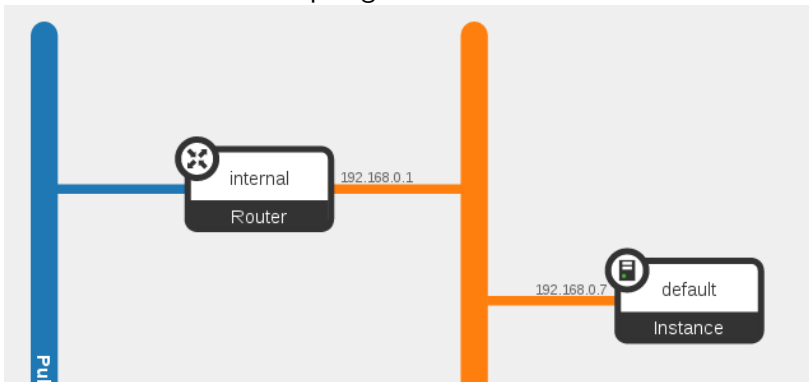
OpenStack Provider – Netzwerk 1/3

```
{"forbidden": {"message": "It is not allowed to create an interface on external network 50ea4b59-42e9-4427-9f00-16362fd0cfd9", "code": 403}}
```

- 1 Neues Netzwerk `internal` anlegen, (z. B. `192.168.0.0/24`).
- 2 Neuen Router anlegen: `router`.
- 3 `PublicNetwork` als Gateway für den Router verwenden.
- 4 Netzwerk als Schnittstelle zum Router hinzufügen.

OpenStack Provider – Netzwerk 2/3

- Resultierende Netzwerktopologie:



OpenStack Provider – Netzwerk 3/3

- neu angelegtes Netzwerk (`internal`) explizit angeben
- Verwendung mehrerer Netzwerke möglich

```
config.vm.provider :openstack do |os|
  os.openstack_auth_url = 'https://api.de-fra1.entercloudsuite.com/v2.0/tokens'
  os.username           = 'christian@berendt.io'
  os.password          = 'password'
  os.tenant_name       = 'christian@berendt.io'
  os.flavor             = 'e1standard.x1'
  os.floating_ip_pool  = 'PublicNetwork'
  os.image              = 'GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64'
  os.networks          = [ 'internal' ]
end
```

OpenStack Provider – Bootstrap, zweiter Versuch 1/2

```
$ vagrant up
Bringing machine 'default' up with 'openstack' provider...
==> default: Finding flavor for server...
==> default: Finding image for server...
==> default: Finding network(s) for server...
==> default: Launching a server with the following settings...
==> default: -- Tenant      : christian@berendt.io
==> default: -- Name       : default
==> default: -- Flavor      : e1standard.x1
==> default: -- FlavorRef   : 10
==> default: -- Image       : GNU/Linux Ubuntu Server 14.04 LTS Trusty Tahr x64
==> default: -- ImageRef    : f0094243-0caa-4f7f-81bd-f306dedb2126
==> default: -- KeyPair     : vagrant-generated-nchcjm66
==> default: -- Network     : 96144df4-12a9-4519-9a1a-814175ccb303
==> default: Waiting for the server to be built...
```

OpenStack Provider – Bootstrap, zweiter Versuch 2/2

```
==> default: Using floating IP 185.48.33.107
==> default: Waiting for SSH to become available...
ssh: connect to host 185.48.33.107 port 22: Connection refused
[...]
==> default: Waiting for SSH to become available...
ssh: connect to host 185.48.33.107 port 22: Connection refused
[...]
==> default: Waiting for SSH to become available...
ssh: connect to host 185.48.33.107 port 22: Connection refused
[...]
==> default: Waiting for SSH to become available...
Connection to 185.48.33.107 closed.
==> default: The server is ready!
==> default: Rsyncing folder: /home/berendt/testing/vagrant-openstack-provider/ => /vagrant
```


OpenStack Provider

```
$ vagrant ssh
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-29-generic x86_64)

* Documentation:  https://help.ubuntu.com/

System information as of Wed May 13 22:36:53 UTC 2015

System load:  0.29           Processes:            78
Usage of /:   3.8% of 19.65GB Users logged in:      0
Memory usage: 11%          IP address for eth0: 192.168.0.7
Swap usage:   0%

Graph this data and manage this system at:
  https://landscape.canonical.com/

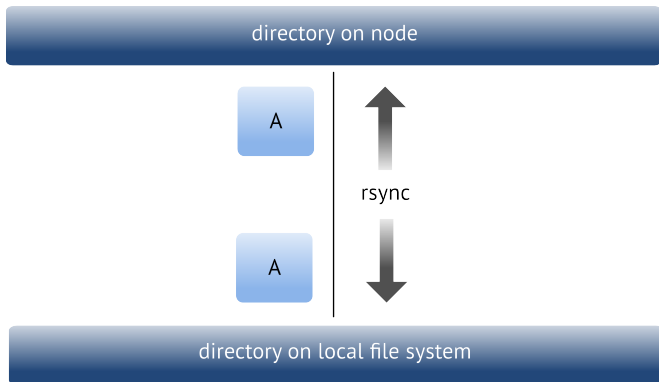
Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Wed May 13 22:36:50 2015 from zitrone.cabtec.net
ubuntu@default:~$
```

OpenStack Provider – Synchronisierte Ordner

==> default: Rsyncing folder: /home/berendt/testing/vagrant-openstack-provider/ => /vagrant



OpenStack Provider – Port Forwarding

```
config.vm.network "forwarded_port", guest: 80, host: 8080
```

- `config.vm.network` Features nicht unterstützt
- `os.security_groups` verwenden, um vordefinierte Security Groups anzuwenden
- Anwenden einer existierenden Security Group (`http`) mit folgendem Eintrag in `Vagrantfile`:

```
os.security_groups = [ 'http' ]
```

OpenStack Provider – Zusätzlicher Speicher

- existierende Volumes anzeigen: `vagrant openstack volume-list`

Id	Name	Size (Go)	Status	Attachment (Instance id and device)
e48e5320-20c0-4bdd-9216-860a301d1108	vagrant	1	available	

- Diese Session verwendet `vagrant`.
- Folgende Zeile zu Vagrantfile hinzufügen:

```
os.volumes = [ 'vagrant' ]
```

Vagrant-Beispiele

- `devstack-vagrant` (openstack-dev project)
- `monasca-vagrant` (stackforge project)
- `packstack-vagrant` (stackforge project)

Vielen Dank für Ihre Aufmerksamkeit!

Bei weiteren Fragen wenden Sie sich bitte an info@b1-systems.de
oder +49 (0)8457 - 931096